



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 10-2003-0020103  
Application Number

출원 년 월 일 : 2003년 03월 31일  
Date of Application MAR 31, 2003

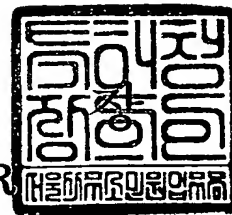
출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2004 년 02 월 03 일

특 허 청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0002
【제출일자】	2003.03.31
【발명의 명칭】	플래시 메모리 액세스 장치 및 방법
【발명의 영문명칭】	ACCESS APPARATUS AND METHOD USING FLASH MEMORY
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	김동진
【대리인코드】	9-1999-000041-4
【포괄위임등록번호】	2002-007585-8
【발명자】	
【성명의 국문표기】	김범수
【성명의 영문표기】	KIM,Bum Soo
【주민등록번호】	690121-1019710
【우편번호】	431-080
【주소】	경기도 안양시 동안구 호계동 1055-1 무궁화아파트 703동 203호
【국적】	KR
【발명자】	
【성명의 국문표기】	권용석
【성명의 영문표기】	KWON,Yong seok
【주민등록번호】	741205-1398915
【우편번호】	139-909
【주소】	서울특별시 노원구 월계1동 동신아파트 5동 803호
【국적】	KR
【발명자】	
【성명의 국문표기】	정태선
【성명의 영문표기】	CHUNG,Tae Sun
【주민등록번호】	710304-1030617

【우편번호】 156-845

【주소】 서울특별시 동작구 상도4동 279-400

【국적】 KR

【심사청구】 청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
김동진 (인)

【수수료】

【기본출원료】	20	면	29,000	원
【가산출원료】	17	면	17,000	원
【우선권주장료】	0	건	0	원
【심사청구료】	17	항	653,000	원
【합계】	699,000 원			

【첨부서류】 1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

본 발명은 플래시 메모리 액세스 장치 및 방법에 관한 것으로, 최소화된 기록 연산과정과 이를 고려한 복구 연산 과정을 통해 반복적인 기록 연산으로 인한 플래시 메모리의 성능 저하를 최소화하며 오류가 발생하더라도 안정적으로 복구 가능하도록 하여 플래시 메모리 시스템의 전체 성능을 높이는 것을 주 목적으로 한다.

이를 위해 본 발명에 따른 플래시 메모리 액세스 장치는 소정의 블록으로 이루어지는 유닛을 단위로 영역이 구분된 플래시 메모리, 플래시 메모리의 특정 논리 블록 번호에 대해 요청되는 기록 연산에서 상기 논리 블록에 대한 이전의 기록 연산이 없을 경우 상기 논리 블록에 대응하는 물리 블록에 데이터와 메타 정보를 기록하고, 이전의 기록 연산이 있을 경우 상기 논리 블록에 대응하는 이전의 물리 블록에 기록된 플래시 메모리 상태 정보를 변경하지 않고 새로운 물리 블록에 상기 논리 블록으로 지정된 데이터와 메타 정보를 기록하는 기록 연산을 수행하는 플래시 메모리 컨트롤러를 포함하여 구성된다.

**【대표도】**

도 3

**【색인어】**

플래시 메모리, 액세스 장치, 플래시 파일 시스템

## 【명세서】

## 【발명의 명칭】

플래시 메모리 액세스 장치 및 방법{ACCESS APPARATUS AND METHOD USING FLASH MEMORY}

## 【도면의 간단한 설명】

도 1은 종래의 플래시 메모리 제어 방법에 따른 연산 과정을 나타내기 위한 플래시 메모리 블록도이다.

도 2의 a는 종래의 플래시 메모리 재사상 제어 방법에 따른 랩 카운트 개념도이다.

도 2의 b는 종래의 플래시 메모리 재사상 제어 방법에 따른 플래시 메모리 블록 구조를 나타낸 구조도이다.

도 2의 c는 상태 정보를 각 블록에 기록하고 오류 발생 시 이후 데이터로 복구하는 과정을 나타낸 플래시 메모리 블록도이다.

도 3은 본 발명의 일 실시 예에 따른 플래시 메모리 액세스 장치를 나타낸 것이다.

도 4는 본 발명의 일 실시 예에 따른 NAND 타입 플래시 메모리의 블록 구조를 나타낸 것이다.

도 5는 본 발명의 일 실시 예에 따른 플래시 메모리의 기록 연산 과정을 나타낸 동작 흐름도이다.

도 6은 본 발명의 일 실시 예에 따른 플래시 메모리의 초기화 과정을 나타낸 동작 흐름도이다.

도 7은 본 발명의 일 실시 예에 따른 플래시 메모리의 재생 과정을 나타낸 동작 흐름도이다.

도 8의 a 내지 c는 본 발명의 일 실시 예에 따른 플래시 메모리 액세스 과정을 나타낸 것이다.

\* 도면의 주요부분에 대한 부호의 설명 \*

100 : 플래시 메모리

110 : 주 영역

130 : 보조 영역

300 : 플래시 메모리 컨트롤러

500 : 프로세서

700 : 램(RAM)

#### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

<16> 본 발명은 플래시 메모리 액세스 장치 및 방법에 관한 것으로, 특히 프로세서로부터 요구되는 기록 연산에 따라 플래시 메모리를 액세스하는 기록 연산 과정을 최소화하며 이에 따른 복구 연산 과정을 통해 오류 발생을 방지함으로써 시스템 오류 발생 시 복구 가능하도록 하면서 플래시 메모리의 성능 저하를 최소화하는 것이 가능한 플래시 메모리 액세스 장치 및 방법에 관한 것이다.

<17> 일반적으로, 플래시 메모리는 비휘발성을 가지면서도 프로그램 가능한 메모리로서 빠른 접근 속도를 가지며 전력을 적게 소비하기 때문에, 이동 단말기, PDA 등의 이동 디바이스(mobile device)를 비롯한 내장형 시스템(embedded system)에서 널리 사용되고 있다.

- <18> 이러한 플래시 메모리는 기존의 RAM이나 비휘발성 저장 장치, 마그네틱 등과 마찬가지로 특정 위치에 저장된 데이터를 임의로 접근할 수 있지만, 데이터를 수정하거나 삭제하는 방법에 있어서는 기존의 저장장치와 다르게 블록(block)을 기본단위로 하여 액세스가 이루어진다.
- <19> 즉, 플래시 메모리에서는 판독/기록 연산 시에 플래시 메모리로부터 한번에 기록된 데이터를 읽어 오거나 데이터를 기록할 수 있는 블록(block)과, 다수의 블록들로 이루어지며 한번의 삭제 연산으로 지울 수 있는 유닛(unit)로 액세스가 이루어지는데, 이 같은 플래시 메모리의 액세스 특성에 따라 데이터를 효율적으로 관리하기 위한 방법으로 블록(또는 섹터) 재사상(re-mapping)기법이 일반적으로 이용되고 있다.
- <20> 블록(또는 섹터) 재사상 기법은 매핑 테이블을 통해 플래시 메모리에 기록되는 특정 데이터에 대한 물리 블록 번호(Physical Block Number:이하, PBN이라 칭함)와 논리 블록 번호(Logical Block Number:이하, LBN이라 칭함)간의 사상 정보를 이용하여 해당 데이터의 수정 또는 삭제로 PBN이 변경되어도 동일한 LBN으로 데이터에 접근할 수 있도록 하는 것이다.
- <21> 이와 같은 기존의 블록 재사상 기법은 프로세서로부터 요구되는 기록/삭제/수정 연산 수행 시 마다 매핑 테이블에 기록된 플래시 메모리의 상태 정보를 수정하여야 하는데, 이로 인한 반복적인 기록 연산에 의해 플래시 메모리의 성능이 저하되는 문제점이 있었다.
- <22> 이러한 문제점을 해결하기 위해 플래시 메모리 제어 방법이 제시되고 있는데, 도 1은 이 같은 플래시 메모리 제어 방법에 따른 연산 과정을 나타내기 위한 플래시 메모리 블록이다.
- <23> 먼저, 앞서 출원된 '국내특허 출원 제 1999-0041835호(발명의 명칭 : 플래시 메모리와 그 제어 방법)'에 따른 것으로, 도 1의 아래 부분은 물리 유닛의 구조를 나타낸 것이다.

- <24> 각 유닛은 헤더(EUH:Erase Unit Header)와 블록 할당 맵(Block Allocation Map:이하, BAM라 칭함)을 가지며 실제 데이터가 저장되는 것으로, 헤더에는 유닛에 대한 여러 메타 정보가 저장되고, BAM에는 특정 유닛에 속한 블록들에 대한 정보(논리 블록 번호, 상태 등)가 기록된다.
- <25> 이러한 플래시 메모리의 특정 LBN(3)에 대한 기록 연산이 요청되면 먼저 LBN-to-LUN 테이블을 이용하여 주어진 LBN(3)에 대한 논리 유닛 번호(Logical Unit Number:이하, LUN라 칭함)(2)를 탐색하고, LUN(2)이 결정되면 LUN-to-PUN 테이블을 이용해 물리 유닛 번호(Physical Unit Number:이하, PUN라 칭함)를 검색한다.
- <26> 이에 따라 PUN(1)이 결정되면, PUN1에 상응하는 BAM을 통해 현재 쓰기 가능한 PBN을 검색한 후 검색된 PBN(1)에 기록 연산이 요구되는 데이터를 기록하고, PBN(1)에 상응하는 BAM의 첫 번째 필드에 LBN(3) 및 상태 정보(유효;valid)를 기록한다.
- <27> 이와 같이, 기록된 LBN(3)에 대한 데이터를 갱신하기 위한 연산이 요구되면, 새로운 PBN을 할당받아 갱신되는 LBN(3)의 데이터를 기록하기 위해 할당된 PBN(4)에 데이터를 기록하고, PBN(4)에 상응하는 BAM의 네 번째 필드에 LBN(3) 및 상태 정보(유효;valid)를 기록한다.
- <28> 그리고, 갱신된 LBN(3)이 기록되었던 이전의 PBN(1)에 기록된 데이터 및 PBN(1)에 상응하는 BAM의 첫 번째 필드에 기록된 LBN(3)를 삭제하고, 상태 정보(삭제:delete)를 기록한다.
- <29> 이러한 기록 및 갱신 연산 과정을 통해 PUN 1에 여유 블록이 존재하지 않는 경우, 플래시 메모리를 보다 효율적으로 사용하기 위하여 PUN 1에 기록된 유효한 데이터만을 새로운 PUN 2에 복사한다.



- <30> 또한, 이러한 문제점을 해결하기 위해 앞서 제안된 것으로 국내특허 출원 제 2000-0059731호(발명의 명칭: 플래시 메모리를 위한 재사상 제어방법 및 그에 따른 플래시 메모리의 구조)는 도 2의 a에 도시한 바와 같이 오류 발생 시 복구를 위하여 블록의 상태 정보를 순환계수(wrap-count)를 사용해 표시(1111:미정상상태, 1110(S0), 1100(S1), 1000(S2):유효한 상태, 0000:삭제)하도록 하여, 순환 계수의 변화 순서를 이용해 이전의 데이터와 새 데이터를 구분해 오류 발생 시 최소의 부분 쓰기 연산으로 시스템을 복구할 수 있도록 하는 것이다.
- <31> 도 2의 b는 이러한 플래시 메모리 재사상 제어방법에 따른 플래시 메모리 구조를 나타낸 것으로, 실질적인 데이터가 기록되는 데이터(data), 논리적인 섹터(블록) 번호(logical sector number; lsn), 순환계수(wrap count; cnt), lsn과 cnt의 에러 정정 코드(ecc\_lsn), 데이터에 대한 에러 정정 코드(ecc\_data)로 구성된다.
- <32> 도 2의 C는 상태 정보를 각 블록에 기록하고 오류 발생 시 이후 데이터로 복구하는 과정을 나타낸 것이다.
- <33> 예를 들어, 다섯 번째 플래시 메모리 블록 이후에 오류가 발생하면, 유효 데이터가 두 가지 존재하고 각각 랩 카운트가 1110과 1100인데, 1110이 1100에 순서적으로 우선하므로 최신 데이터로 복구하는 정책을 쓴다면 왼쪽 데이터가 삭제된다.
- <34> 이와 같이, 종래의 플래시 메모리 제어 방법의 경우에도 반복적인 기록 연산에 의한 플래시 메모리의 시스템 성능을 저하를 방지하기 위한 방법으로는 역부족이었다.
- <35> 따라서, 이러한 쓰기 연산의 횟수를 줄임으로써 플래시 메모리 시스템의 성능을 향상시킬 수 있는 플래시 메모리 액세스 방법이 요구되고 있다.

**【발명이 이루고자 하는 기술적 과제】**

- <36> 본 발명은 상기한 문제점들을 해결하기 위해 안출된 것으로, 프로세서로부터 요구되는 갱신 연산 과정에서 이전의 PBN에 변경되는 플래시 메모리 상태 정보를 기록하는 기록 연산과정을 생략함으로써 기록 연산 과정을 최소화하는 것을 목적으로 한다.
- <37> 또한, 본 발명의 다른 목적은 최소화된 기록 연산 과정에 따라 플래시 메모리의 초기화 과정에서 특정 LBN의 데이터가 하나 이상 존재하더라도 우선 순위를 통해 최신 데이터를 판별하여 처리하는 복구 연산 과정을 제공하는 것이다.
- <38> 또한, 본 발명의 또 다른 목적은 최소화된 기록 연산과정과 이를 고려한 복구 연산 과정을 통해 반복적인 기록 연산으로 인한 플래시 메모리의 성능 저하를 최소화하며 오류가 발생하더라도 안정적으로 복구 가능하도록 하여 플래시 메모리 시스템의 전체 성능을 높이는 것이다.

**【발명의 구성 및 작용】**

- <39> 본 발명은 복구 연산 과정을 고려한 기록 연산에 따라 프로세서로부터 요구되는 갱신 연산 과정에서 이전의 PBN에 변경되는 플래시 메모리 상태 정보를 기록하는 기록 연산과정을 생략하고, 플래시 메모리의 초기화 과정에서 특정 LBN의 데이터가 하나 이상 존재하더라도 복구 연산 과정을 통해 최신 데이터를 판별하여 처리한다.
- <40> 복구 연산 과정을 통해 최신 데이터를 판별이 가능하도록 하기 위하여 초기화 과정에서는 플래시 메모리의 하위 블록에 우선순위를 부여하고, 이를 기준으로 각 블록을 스캐닝하면서 각 블록의 LBN에 대한 데이터 기록 여부를 블록 할당 테이블의 상응하는 LBN에 기록한다.

- <41> 또한, 플래시 메모리의 소정 유닛에 존재하는 유틸 데이터를 새로운 유닛으로 이동시키는 메모리 재생 과정을 통해서도 복구 연산 과정을 수행하여 최신 데이터를 판별하고, 이전의 데이터를 삭제 처리한다.
- <42> 이하, 본 발명에 따른 플래시 메모리 액세스 장치 및 방법을 첨부된 도면을 참조하여 상세히 설명한다.
- <43> 본 발명에 따른 플래시 메모리 액세스 장치는 소정의 블록으로 이루어지는 유닛을 단위로 영역이 구분된 플래시 메모리, 플래시 메모리의 특정 논리 블록 번호에 대해 요청되는 기록 연산에서 상기 논리 블록에 대한 이전의 기록 연산이 없을 경우 상기 논리 블록에 대응하는 물리 블록에 데이터와 메타 정보를 기록하고, 이전의 기록 연산이 있을 경우 상기 논리 블록에 대응하는 이전의 물리 블록에 기록된 플래시 메모리 상태 정보를 변경하지 않고 새로운 물리 블록에 상기 논리 블록으로 지정된 데이터와 메타 정보를 기록하는 기록 연산을 수행하는 플래시 메모리 컨트롤러를 포함하는 것을 특징으로 한다.
- <44> 본 발명에 따른 플래시 메모리 액세스 방법은 프로세서로부터 특정 논리 블록 번호에 대한 기록 연산이 요구되면 플래시 메모리에 접근하여 기록 연산을 위해 현재 쓰기 가능한 물리 블록을 검색하는 단계; 및 논리 블록 번호에 대한 이전의 기록 연산이 없을 경우 상기 논리 블록에 대응하는 물리 블록에 데이터와 메타 정보를 기록하고, 이전의 기록 연산이 있을 경우 상기 논리 블록에 대응하는 이전의 물리 블록에 기록된 플래시 메모리 상태 정보를 변경하지 않고 상기 논리 블록에 대응하는 새로운 물리 블록에 데이터와 메타 정보를 기록하는 단계를 포함하는 것을 특징으로 한다.
- <45> 우선, 본 발명에 따른 플래시 메모리의 구조 및 그에 따른 플래시 메모리 액세스 장치의 구성과 동작을 첨부된 도면을 참조하여 일 실시 예로 상세히 설명한다.

- <46> 도 3은 본 발명의 일 실시 예에 따른 플래시 메모리 액세스 장치를 나타낸 것이다.
- <47> 도 3에 도시된 바와 같이, 본 발명에 따른 플래시 메모리 액세스 장치는 플래시 메모리(100), 플래시 메모리(100)에 접근하여 지정되는 논리 블록의 데이터 및 메타 정보를 쓰기 가능한 물리 블록에 기록하는 기록 및 갱신 연산(이하, 기록 연산으로 총칭함)에 상응하는 동작 제어를 수행하는 플래시 메모리 컨트롤러(300)로 구성되며, 플래시 메모리 컨트롤러(300)를 통해 플래시 메모리(100)에 기록된 데이터를 판독하거나 데이터를 기록하기 위한 제어 명령어를 발생시키는 시스템의 프로세서(500), 시스템 성능을 높이기 위하여 플래시 메모리(100)에 저장되어 있는 각종 정보(일 예로, 논리-물리 유닛간의 매핑 정보, 각 논리 블록에 대한 액세스 정보, 각 유닛의 논리-물리 블록간의 매핑 정보)를 캐쉬(cache)하는 램(700)을 더 포함한다.
- <48> 또한, 플래시 메모리 컨트롤러(300)는 초기화 과정이나 유효 데이터를 새로운 유닛으로 이동시키는 메모리 재생 과정에서 특정 LBN의 데이터가 하나 이상 존재하면 스캐닝 과정에서 주어지는 우선 순위를 통해 최근 데이터를 판별하여 이외의 데이터에 대한 플래시 메모리 상태 정보를 삭제(D) 처리하는 복구 연산 과정을 수행한다.
- <49> 초기화 및 메모리 재생 과정은 플래시 메모리(100)의 하위 블록을 기준으로 각 블록에 기록된 LBN을 스캐닝하는 과정과, 스캐닝 과정에서 얻어지는 각 LBN에 데이터가 존재하는지 여부에 대한 조사 결과 값을 표시하는 '0' 또는 '1'을 블록 할당 테이블(Block Allocation Table:이하, BAT라 칭함)에 기록하는 과정을 포함한다.
- <50> BAT는 플래시 메모리(100)의 효율적인 액세스를 위하여 램에 저장되는 것으로, 복구 연산 과정에서 특정 LBN의 데이터 가운데 최근 데이터를 판별할 수 있도록 하기 위해 스캐닝 과정에서 얻어지는 각 LBN의 데이터 기록 여부를 표시하기 위한 테이블이다.

- <51> 일 예로, 플래시 메모리(100)의 물리 블록을 스캐닝하고 스캐닝되는 물리 블록(100)의 데이터 존재 여부를 BAT의 해당 LBN에 기록하는 플래시 메모리 초기화 과정에서, 선택된 블록의 LBN에 상응하는 BAT의 상응하는 LBN 필드에 '1'이 기록되어 있으면 검색된 물리 블록에 기록된 데이터의 LBN이 이전의 스캐닝 과정에서 검색된 LBN 값이고, 검색된 물리 블록의 데이터는 이전의 물리 블록에 기록된 데이터보다 우선 순위가 낮기 때문에 검색된 물리 블록의 플래시 메모리 상태 정보를 삭제(D)로 변경한다.
- <52> 플래시 메모리(100)는 삭제 단위인 여러 개의 유닛으로 구성되고, 각 유닛은 여러 개의 블록으로 분할되며 플래시 메모리(100)에 대한 읽기나 쓰기 연산은 블록 단위로 이루어 진다.
- <53> 도 4는 본 발명의 일 실시 예에 따른 NAND 타입의 플래시 메모리 블록 구조를 나타낸 것이다.
- <54> 도 4에 도시된 바와 같이, NAND 타입의 플래시 메모리(100)는 주 영역(main area;110)과 보조 영역(spare area;130)로 나누어 지는데, 주 영역(110)에는 데이터가 저장되고, 보조 영역(130)에는 해당 블록에 대한 메타 정보가 저장된다.
- <55> 일 예로, 메타 정보는 논리 블록 번호(LBN), 플래시 메모리 상태 정보(유효(valid), 삭제(delete), 무효(invalid))가 기록되는 각 분할 영역으로 이루어지며, 에러 검출 코드(Error Correction Code;ECC)가 기록되는 영역을 더 포함한다.
- <56> NOR 타입의 플래시 메모리(100)인 경우에는 블록에 대한 상태 정보가 각 유닛 앞 부분의 하나 혹은 그 이상의 블록에 저장될 수 있다.

- <57>       참고로, 전술한 본 발명의 일 실시 예에 따른 플래시 메모리 액세스 장치는 각 모듈이 모두 하드웨어로 구성되거나, 일부 모듈이 소프트웨어로 구성되거나, 또는 전체 모듈이 소프트웨어로 구성될 수 있다.
- <58>       따라서, 본 발명의 일 실시 예에 따른 플래시 메모리 액세스 장치가 하드웨어 또는 소프트웨어로 구성되는 것은 본 발명의 사상을 벗어나지 않으며, 본 발명의 사상에서 벗어나지 않으면서 소프트웨어 및/또는 하드웨어로 구성됨에 따른 수정과 변경이 부가될 수 있음은 자명하다.
- <59>       이와 같은 본 발명에 따른 플래시 메모리 및 플래시 메모리 액세스 장치를 이용하여 플래시 메모리를 액세스하는 방법을 첨부된 도면을 참조하여 일 실시 예로 상세히 설명한다.
- <60>       본 발명에 따른 플래시 메모리(100)의 액세스 과정을 설명함에 있어서, 판독 연산 과정은 기존의 처리 과정과 동일하므로 이에 따른 설명은 생략한다.
- <61>       따라서, 먼저 본 발명에 따른 복구 연산 과정을 고려한 기록 연산 과정을 설명하고, 초기화 과정에서 오류 복구를 위한 기록 연산을 수행하는 오류 복구 연산 과정을 설명한다.
- <62>       또한, 플래시 메모리(100)에 기록된 데이터 가운데 유효한 데이터 만을 새로운 유닛으로 이동시키는 과정에서 오류 복구를 위한 기록 연산을 수행하는 메모리 재생(reclaim) 과정을 설명한다.
- <63>       1. 기록 연산 과정
- <64>       기록 연산 과정은 프로세서(500)로부터 요구되는 기록 연산에 따라 지정된 논리 블록에 대한 데이터를 플래시 메모리(100)의 특정 물리 블록에 기록하기 위하여 사용 가능한 물리 블록을 검색하고, 데이터 및 이에 따른 메타 정보를 물리 블록에 기록하는 것이다.

- <65> 도 5는 본 발명의 일 실시 예에 따른 플래시 메모리의 기록 연산 과정을 나타낸 동작 흐름도이다.
- <66> 도 5에 도시된 바와 같이, 먼저, 플래시 메모리 컨트롤러(300)에서는 기록 커맨드(write command)와 함께 논리 블록 주소가 입력되면 데이터가 기록될 물리 블록을 결정하기 위해, LBN-LUN 테이블을 통해 지정된 LBN로 LUN을 검색하고(S1), 검색된 LUN를 이용해 논리-물리 유닛간의 매핑 테이블(LUN-PUN 테이블)에서 PUN를 검색한다(S2).
- <67> 그 다음, 검색된 물리 유닛 번호의 물리 블록 가운데 현재 쓰기 가능한 물리 블록이 존재하는지 여부를 검색하기 위하여 BAM을 조사하고(S3), 검색된 물리 유닛 번호를 구성하는 물리 블록 중 자유블록이 존재하는지 여부를 판별한다(S4).
- <68> 판별결과 자유 블록이 존재하지 않으면 해당 물리 유닛에 기록된 데이터들 가운데 특정 논리 블록의 유효 데이터만을 새로운 유닛으로 이동시키는 메모리 재생(reclaim)을 수행한 후 LUN-PUN 테이블을 통해 PUN을 검색하는 루틴으로 리턴하고(S5), 판별결과 자유 블록이 존재하면 검색된 자유 블록에 데이터 및 메타 정보를 기록한다(S6).
- <69> 데이터 및 메타 정보는 해당 데이터가 안정적으로 기록되었음을 보장하기 위해, 데이터를 블록의 주 영역에 기록하고, 메타 정보인 지정된 논리 블록 번호를 블록의 보조 영역에 기록한다.
- <70> 이러한 과정을 통해 데이터 및 논리 블록 번호에 대한 기록 연산이 완료되면, 기록 연산에 따른 플래시 메모리(100)의 상태 정보(유효)를 블록의 보조 영역에 기록한다.
- <71> 만일, 블록 단위의 기록 연산에 대한 원자성(atomicity)이 보장된다면, 데이터 및 메타 정보는 동시에 기록될 수 있다.

<72> 2. 초기화 과정

<73> 초기화 과정은 프로세서(500)로부터 요구되는 연산 동작에 따라 플래시 메모리(100)를 효율적으로 액세스하기 위하여 플래시 메모리 컨트롤러(300)에서 플래시 메모리(100)에 기록된 사상 정보를 로딩하는 것이다.

<74> 이하, 본 발명에 따라 일 실시 예로 설명하는 초기화 과정은 갑작스런 전원 중단으로 인해 이루어지는 초기화 과정으로 정의되는 것으로, 초기화 과정에서 일어나는 갑작스런 전원 중단으로 인한 오류를 복구하는 오류 복구과정을 나타낸 것이다.

<75> 그러나, 본 발명은 갑작스런 전원 중단에 의한 초기화 과정에 한정되지 않고 모든 플래시 메모리(100) 초기화 과정에 적용 가능하다.

<76> 도 6은 본 발명의 일 실시 예에 따른 플래시 메모리의 초기화 과정을 나타낸 동작 흐름도이다.

<77> 도 6에 도시된 바와 같이, 먼저, 플래시 메모리 컨트롤러(300)에서는 플래시 메모리(100)의 헤더에 기록되는 플래시 메모리 전체 정보를 로딩한 후, 플래시 메모리(100)를 스캐닝하기 위해 가장 최근 액세스된 유닛의 가장 하위 물리 블록을 선택하고(S11), 선택된 물리 블록을 우선 순위로 하여 플래시 메모리(100)를 구성하는 물리 블록의 스캐닝을 시작한다.

<78> 이에 따라, 선택된 물리 블록에서 LBN을 독출하고(S12), 독출된 LBN을 이용해 BAT에서 상응하는 LBN 필드를 검색한다(S13).

<79> 그 다음, 검색된 LBN 필드에 기록된 상태 값이 '1'인지 여부를 판별하여(S14), 판별결과 상태 값이 '1'이면 BAT의 LBN 상태 값에 '1'을 기록하고(S15), 선택된 물리 블록이 해당 유닛의 첫 번째 블록인지 여부를 판별한다(S16).



- <80> 판별 결과 첫 번째 블록이 아니면 한 단계 상위 블록으로 이동하여 선택하고, 선택된 블록에서 LBN 상태 값을 독출하는 루틴으로 리턴한다.
- <81> 초기 첫 번째 블록에 대한 스캐닝 과정이 수행된 후, BAT에서 검색된 LBN 상태 값이 '1' 인지 여부에 대한 판별 결과 LBN 상태 값이 '1'이면 해당 블록의 플래시 메모리 상태 정보에 삭제(delete) 표시한다(S18).
- <82> 이러한 과정을 통해 스캐닝 과정에서 BAT에 기록된 상응하는 LBN의 상태 값이 '1'로 검색되는 물리 블록이 검색되면, 검색된 물리 블록에 기록된 LBN의 데이터는 이전에 검색된 블록의 LBN 데이터보다 우선순위가 낮으므로 상태 정보를 삭제 표시하여 동일한 LBN을 갖는 유효 데이터로 인한 오류를 방지하는 것이다.
- <83> 3. 재생(reclaim) 과정
- <84> 재생 과정은 프로세서로부터 기록 연산이 요구되는 데이터를 기록하기 위한 현재 쓰기 가능한 자유 블록이 플래시 메모리의 소정 유닛에 존재하지 않는 경우 해당 유닛에 기록된 유효 데이터만을 새로운 유닛에 복사하는 것이다.
- <85> 도 7은 본 발명의 일 실시 예에 따른 플래시 메모리의 재생 과정을 나타낸 동작 흐름도이다.
- <86> 도 7에 도시된 바와 같이, 먼저 플래시 메모리 컨트롤러(300)에서는 기록된 유효 데이터의 이동이 요구되는 소스 유닛과 유효 데이터를 이동하고자 하는 새로운 유닛인 이동 유닛을 결정한다(S21).
- <87> 그 다음, 소스 유닛의 가장 하위 블록을 선택하고(S22), 선택된 블록의 LBN을 독출한다(S23).

- <88> 그 다음, 독출된 LBN을 이용해 BAT에서 상응하는 LBN을 검색하여 검색된 LBN의 상태 값이 '1'인지 여부를 판별한다(S24).
- <89> 판별결과 LBN의 상태 값이 '1'이 아니면 BAT의 LBN 상태 값을 '1'로 설정하고(S25), LBN의 데이터를 이동 유닛에 복사한다(S26).
- <90> 판별결과 이전에 상응하는 LBN의 데이터가 검색되어 LBN의 상태 값이 '1'로 판별되거나, LBN의 데이터를 이동 유닛에 복사하면, 현재 선택된 블록이 소스 유닛의 첫 번째 블록인지 여부를 판별한다(S27).
- <91> 판별결과 소스 유닛의 첫 번째 블록이 아니면 한 단계 상위 블록으로 이동하여(S28) 해당 블록을 선택하고 선택된 블록의 LBN을 독출하는 루틴으로 리턴한다.
- <92> 판별결과 소스 유닛의 첫 번째 블록이면 소스 유닛의 모든 블록을 삭제하고(S29), 이에 따라 LUN-PUN 사상 정보를 변경한다(S30).
- <93> 이러한 소스 유닛에 기록된 유휴 데이터를 이동 유닛으로 복사하는 과정을 통해 소스 유닛에 존재하는 특정 논리 블록에 대한 다수의 유휴 데이터 가운데 하위 블록에 존재하는 데이터만을 유휴 데이터로 이동 유닛에 복사함으로써, 본 발명에 따른 오류 복구 과정을 수행한다.
- <94> 이와 같이 이루어지는 본 발명에 따른 플래시 메모리 액세스 방법을 첨부된 도면을 참조하여 일 실시 예로 상세히 설명한다.
- <95> 도 8의 a 내지 c는 본 발명의 일 실시 예에 따른 플래시 메모리 액세스 과정을 나타낸 것이다.
- <96> 도 8의 a에 도시된 바와 같이, 플래시 메모리 컨트롤러(300)에서는 프로세서(500)로부터 특정 LBN(3)에 대한 기록 연산이 요구되면 데이터 기록이 가능한 플래시 메모리(100)의 물리

블록에 접근하기 위해, 우선 LBN-LUN 매핑 테이블을 참조하여 기록 연산이 요구되는 LBN(3)이 속하는 LUN을 탐색한다.

<97> LBN-LUN 매핑 테이블을 통해 LUN이 결정되면, LUN-PUN 매핑 테이블을 참조하여 LUN에 대한 PUN을 결정하고, 결정된 PUN의 BAM에서 프로세서(500)에 의해 지정되는 LBN(3) 데이터의 기록이 가능한 물리 블록의 위치를 탐색한다.

<98> 이 과정을 통해 현재 쓰기 가능한 PBN이 검색되면, 플래시 메모리 컨트롤러(300)에서는 검색된 PBN(1)의 주 영역에는 데이터를 기록하고, 보조 영역에는 프로세서에 의해 지정되는 LBN(3)을 기록한다(①).

<99> 이러한 데이터 및 LBN(3) 기록 과정이 완료되면, 프로세서(500)로부터 요구되는 기록 연산이 안정적으로 완료되었으며 이에 따라 해당 블록에 데이터가 기록된 상태임을 표시하도록, 보조 영역의 플래시 메모리(100) 상태 정보를 유효(V)로 기록한다(②).

<100> 이러한 기록 연산이 수행되는 과정에서 프로세서(500)로부터 플래시 메모리의 PBN(1)에 기록된 LBN(3) 데이터의 갱신을 위한 기록 연산이 요구되면, 도 8의 b에 도시된 바와 같이, 플래시 메모리 컨트롤러(300)에서는 갱신되는 데이터를 기록하기 위해 현재 쓰기 가능한 PBN을 검색한다.

<101> 검색 결과 특정 PBN(4)이 자유블록으로 검색되면, 검색된 PBN(4)의 주 영역에는 갱신되는 데이터를 기록하고, 보조 영역에는 할당된 LBN(3)을 기록한다(①).

<102> 그 다음, 데이터 및 LBN(3)에 대한 기록 연산이 완료되면, 프로세서(500)로부터 요구되는 기록 연산이 안정적으로 완료되었으며 이에 따라 PBN(4)에 데이터가 기록된 상태임을 표시하도록 보조영역의 플래시 메모리(100) 상태 정보를 유효(V)로 기록한다(②).

- <103> 이러한 기록 연산이 수행되는 플래시 메모리(100)에서 갑작스런 전원 중단으로 인해 시스템 작동이 중단되고 이후 전원이 인가되면, 플래시 메모리 컨트롤러(300)에서는 플래시 메모리 초기화를 위한 액세스 과정을 수행한다.
- <104> 먼저, 플래시 메모리 컨트롤러(300)에서는 플래시 메모리(100)에서 가장 최근에 액세스된 블록에 접근하기 위해 해당 블록을 포함하는 유닛의 가장 하위 블록인 PBN(4)을 선택한다.
- <105> 그 다음, 선택된 PBN(4)의 보조 영역에 기록된 LBN(3)을 독출하고, 독출된 LBN(3)을 이용해 BAT에서 상응하는 LBN(3) 필드에 기록된 상태 값을 조사한다.
- <106> 조사 결과에 따라 BAT에 기록된 LBN(3)에 대한 상태 값이 '1'이 아니면 BAT의 상응하는 LBN(3) 필드에 상태 값을 '1'로 표시한 후 선택된 PBN(4)이 유닛의 첫 번째 블록인지 여부를 판별하여 한 단위 상위 블록으로 이동하면서 유닛의 각 블록에 대한 상태를 검사하는 스캐닝 동작을 수행한다.
- <107> 이러한 스캐닝 동작 과정에서 BAT에 기록된 LBN에 대한 상태 값이 '1'이면, 이전의 스캐닝 과정에서 유효 데이터를 갖는 동일한 LBN이 검색된 것으로 스캐닝 우선 순위에 따라 현재 선택된 해당 블록의 보조 영역에 기록된 플래시 메모리(100) 상태 정보에 삭제(D) 표시한다.
- <108> 즉, 도 8의 c에 도시한 바와 같이, 특정 LBN(3)에 대한 갱신 데이터가 기록된 PBN(4)이 가장 최근 액세스 되었다면, 플래시 메모리 스캐닝 과정에서 4번째 블록을 유닛의 하위 블록으로 선택하여 LBN(3)을 독출한다(①).
- <109> 독출되는 LBN(3)을 이용해 BAT에서 상응하는 LBN(3)에 기록된 상태 값을 조사하는데, LBN(3)이 가장 최근에 액세스된 하위 블록이므로, 상태 값이 기록되지 않아 '0'이므로 BAT의 상응하는 LBN(3) 필드에 상태 값을 '1'로 기록한다(②).

- <110> 그 다음, 선택된 블록이 유닛의 첫 번째 블록인지 여부를 판별하는데, 선택된 블록은 유닛의 하위 블록이므로 한 단계 상위 블록을 선택한다.
- <111> 이에 따라 PBN(3)이 선택되고(③), 선택된 PBN(3)을 검색하여 얻어지는 LBN(5)를 이용해 BAT의 상응하는 LBN(5) 필드에 기록된 상태 값을 조사한다.
- <112> 조사 결과에 따라, 선택된 PBN(3)에 기록된 데이터의 LBN(5)에 상응하는 BAT의 LBN(5) 필드에는 상태 값이 '1'로 기록되고(④), 현재 선택된 PBN(3)이 첫 번째 블록이 아니므로 한 단계 상위 블록인 PBN(2)가 선택된다.
- <113> 그 다음, 선택된 PBN(2)에 기록된 LBN(4)을 독출하고(⑤), 독출된 LBN(4)에 상응하는 BAT의 LBN(4) 필드에 기록된 상태 값을 검사한 후 상태 값을 '1'로 기록한다(⑥).
- <114> 이러한 스캐닝 과정을 통해, 다음으로 PBN(1)이 선택되면(⑦), 선택된 PBN(1)에 기록된 LBN(3)을 독출하여 BAT의 상응하는 LBN(3) 필드에 기록된 상태 값을 조사한다(⑧).
- <115> 이 과정에서 BAT의 상응하는 LBN(3) 필드에 이전의 스캐닝 과정에서 기록된 정보로 인해 상태 값이 '1'로 기록되어 있으므로, 스캐닝 우선 순위에 따라 이전의 LBN(3)을 포함하는 PBN(4)은 유효한 데이터가 기록된 블록으로 결정되고, PBN(1)은 무효 블록으로 결정된다.
- <116> 이에 따라 PBN(1)의 보조 영역에 기록된 플래시 메모리 상태 정보는 삭제(D)로 변경된다(⑨).
- <117> 이와 같은 스캐닝 과정을 통해 본 발명의 오류 복구 과정을 고려한 최소화된 기록 횟수를 갖는 기록 연산에 의해 특정 논리 블록에 대해 존재하는 유효 데이터들 가운데 최신 데이터를 가려내고, 오류 복구할 수 있다.
- <118> 본 발명은 도면에 도시된 일 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하

는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

**【발명의 효과】**

- <119> 본 발명에 따르면, 플래시 메모리의 특정 논리 블록에 대한 데이터의 갱신 시에 요구되는 기록 연산의 횟수가 최소화되어 플래시 메모리 시스템의 성능이 최대화될 수 있다.
- <120> 또한, 플래시 메모리의 초기화 과정에서 특정 논리 주소의 데이터에 대한 우선 순위를 통해 유효 데이터를 결정함으로써 갑작스런 전원 중단으로 인해 오류가 발생하더라도 안전한 복구가 가능한 효과를 제공한다.

**【특허청구범위】****【청구항 1】**

소정의 블록으로 이루어지는 유닛을 단위로 영역이 구분된 플래시 메모리,

상기 플래시 메모리의 특정 논리 블록 번호에 대해 요청되는 기록 연산에서 상기 논리 블록에 대한 이전의 기록 연산이 없을 경우 상기 논리 블록에 대응하는 물리 블록에 데이터와 메타 정보를 기록하고, 이전의 기록 연산이 있을 경우 상기 논리 블록에 상응하는 이전의 물리 블록에 기록된 플래시 메모리 상태 정보를 변경하지 않고 새로운 물리 블록에 상기 논리 블록으로 지정된 데이터와 메타 정보를 기록하는 기록 연산을 수행하는 플래시 메모리 컨트롤러를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 2】**

제 1항에 있어서,

상기 논리 블록의 데이터를 먼저 기록한 후 상기 메타 정보를 기록하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 3】**

제 1항에 있어서,

상기 논리 블록의 데이터와 메타 정보를 동시에 기록하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 4】**

제 1항에 있어서,

상기 메타 정보는

상기 논리 블록 번호와, 상기 물리 블록에 대한 상태를 유효, 삭제 또는 무효로 표시하는 플래시 메모리 상태 정보를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 5】**

제 1항에 있어서,

상기 플래시 메모리 컨트롤러는

상기 기록 연산에 따라 존재하게 되는 상기 논리 블록 번호에 대한 하나 이상의 물리 블록들을 스캐닝 과정에서 감지하고, 감지된 물리 블록들 가운데 상기 논리 블록에 대한 유효 블록을 판별하여 오류를 복구하는 복구 연산을 수행하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 6】**

제 5항에 있어서,

상기 스캐닝 과정은

상기 플래시 메모리를 가장 최근 액세스된 하위 블록을 기준으로 조사하여 각 물리 블록의 논리 블록 번호를 독출하고, 상기 독출된 논리 블록 번호에 상응하는 블록 할당 테이블의 논리 블록 번호 필드를 검사하는 과정으로 이루어지는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 7】**

제 6항에 있어서,

상기 블록 할당 테이블의 논리 블록 번호 필드를 검사하는 과정은



상기 논리 블록 번호 필드에 기록된 상태 값이 '0'이면 '1'로 기록하고, 상기 상태 값이 '1'이면 상기 스캐닝에서 이전의 물리 블록을 통해 상기 논리 블록 번호가 검색된 것으로 감지하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 8】**

제 5항에 있어서,

상기 유효 블록을 판별하여 오류를 복구하는 것은

상기 감지된 물리 블록들 가운데 스캐닝 과정에서 설정되는 우선 순위에 따라 상기 논리 블록 번호에 대해 가장 최근 액세스된 물리 블록을 유효 블록으로 결정하고, 이외의 물리 블록에 기록된 플래시 메모리 상태 정보를 삭제로 기록하는 과정으로 이루어지는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 9】**

제 5항에 있어서,

상기 복구 연산 과정은

상기 플래시 메모리를 초기화하는 과정에서 이루어지는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 10】**

제 5항에 있어서,

상기 복구 연산 과정은

상기 플래시 메모리의 소정 유닛에 기록된 데이터를 새로운 유닛으로 이동하는 플래시 메모리 재생 과정에서 이루어지는 것을 특징으로 하는 플래시 메모리 액세스 장치.

**【청구항 11】**

프로세서로부터 특정 논리 블록 번호에 대한 기록 연산이 요구되면 플래시 메모리에 접근하여 기록 연산을 위해 현재 쓰기 가능한 물리 블록을 검색하는 단계; 및

상기 논리 블록 번호에 대한 이전의 기록 연산이 없을 경우 상기 논리 블록에 대응하는 물리 블록에 데이터와 메타 정보를 기록하고, 이전의 기록 연산이 있을 경우 상기 논리 블록에 대응하는 이전의 물리 블록에 기록된 플래시 메모리 상태 정보를 변경하지 않고 상기 논리 블록에 대응하는 새로운 물리 블록에 데이터와 메타 정보를 기록하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

**【청구항 12】**

제 11항에 있어서,

상기 데이터를 먼저 기록한 후 상기 메타 정보를 기록하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

**【청구항 13】**

제 11항에 있어서,

상기 데이터와 메타 정보를 동시에 기록하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

**【청구항 14】**

제 11항에 있어서,

상기 메타 정보는

상기 논리 블록 번호와, 상기 물리 블록에 대한 상태를 유효, 삭제 또는 무효로 표시하는 플래시 메모리 상태 정보를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 15】

제 11항에 있어서,

상기 기록 연산에 따라 존재하게 되는 상기 논리 블록 번호에 대한 하나 이상의 물리 블록들을 스캐닝 과정에서 감지하고, 감지된 물리 블록들 가운데 상기 논리 블록에 대한 유효 블록을 판별하여 오류를 복구하는 복구 연산 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 16】

제 15항에 있어서,

상기 스캐닝 과정은

상기 플래시 메모리를 가장 최근 액세스된 하위 블록을 기준으로 조사하여 각 물리 블록의 논리 블록 번호를 독출하고, 상기 독출된 논리 블록 번호에 상응하는 블록 할당 테이블의 논리 블록 번호 필드를 검사하는 단계; 및

상기 논리 블록 번호 필드에 기록된 상태 값이 '0'이면 '1'로 기록하고, 상기 상태 값이 '1'이면 상기 스캐닝에서 이전의 물리 블록을 통해 상기 논리 블록 번호가 검색된 것으로 감지하는 단계로 이루어지는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 17】

제 15항에 있어서,

상기 복구 연산 단계는

상기 플래시 메모리의 소정 유닛에 기록된 데이터를 새로운 유닛으로 이동하는 플래시 메모리 재생 과정에서 감지되는 특정 논리 블록 번호에 대한 데이터들 가운데 최신 데이터를 판별하여 오류를 복구하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【도면】

【도 1】

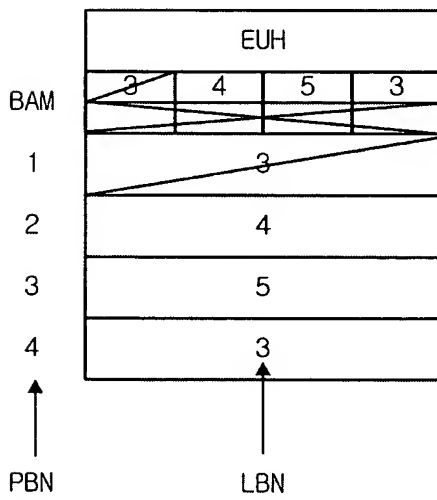
LBN-to-LUN

LBN	LUN
1	1
2	1
3	2
4	2
5	2
.....	.....

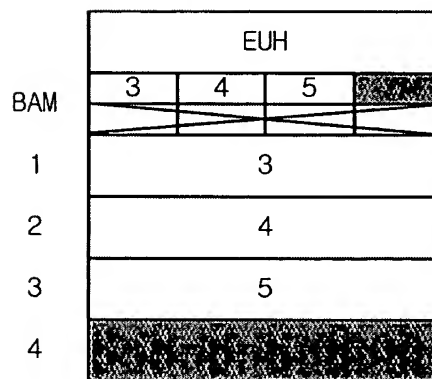
LUN-to-PUN

LUN	PUN
1	2
2	1
.....	.....

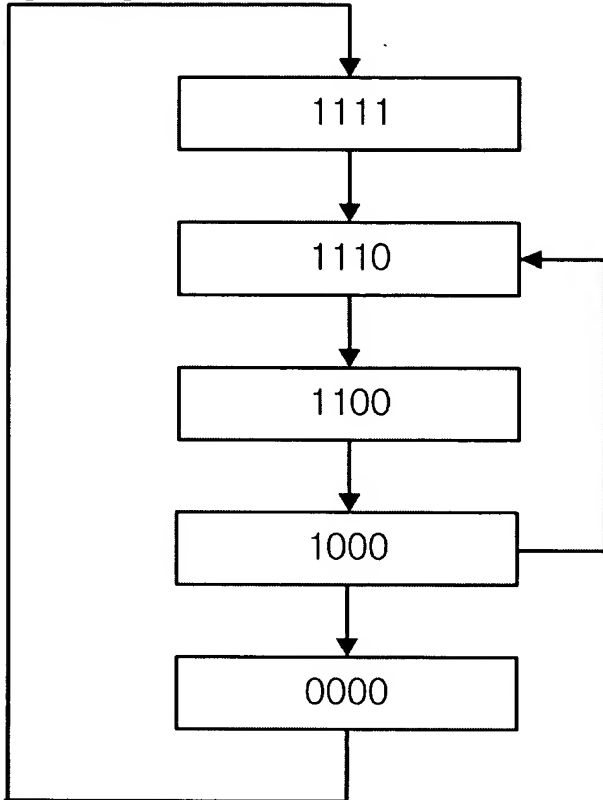
PUN 1



PUN 2



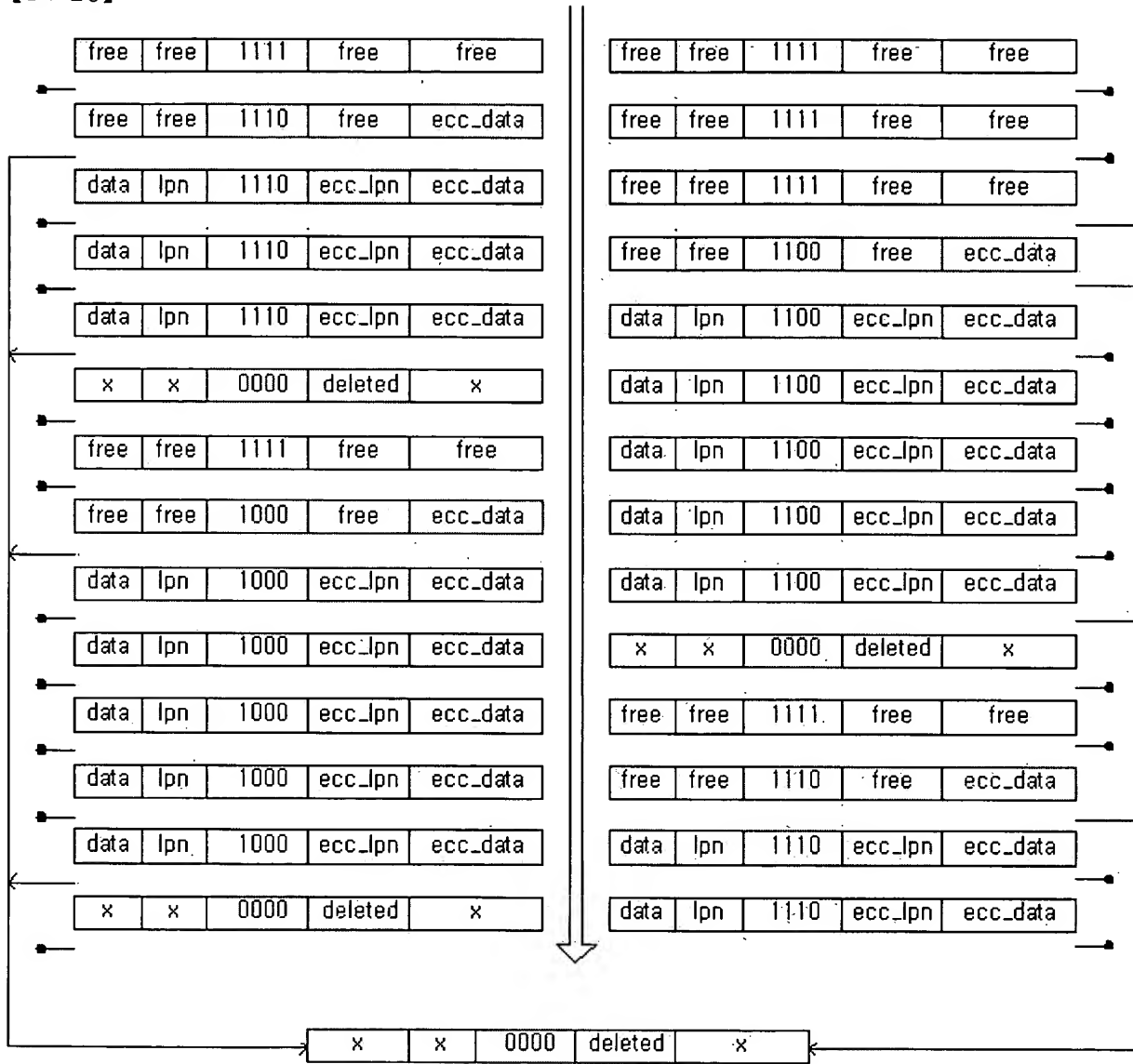
【도 2a】



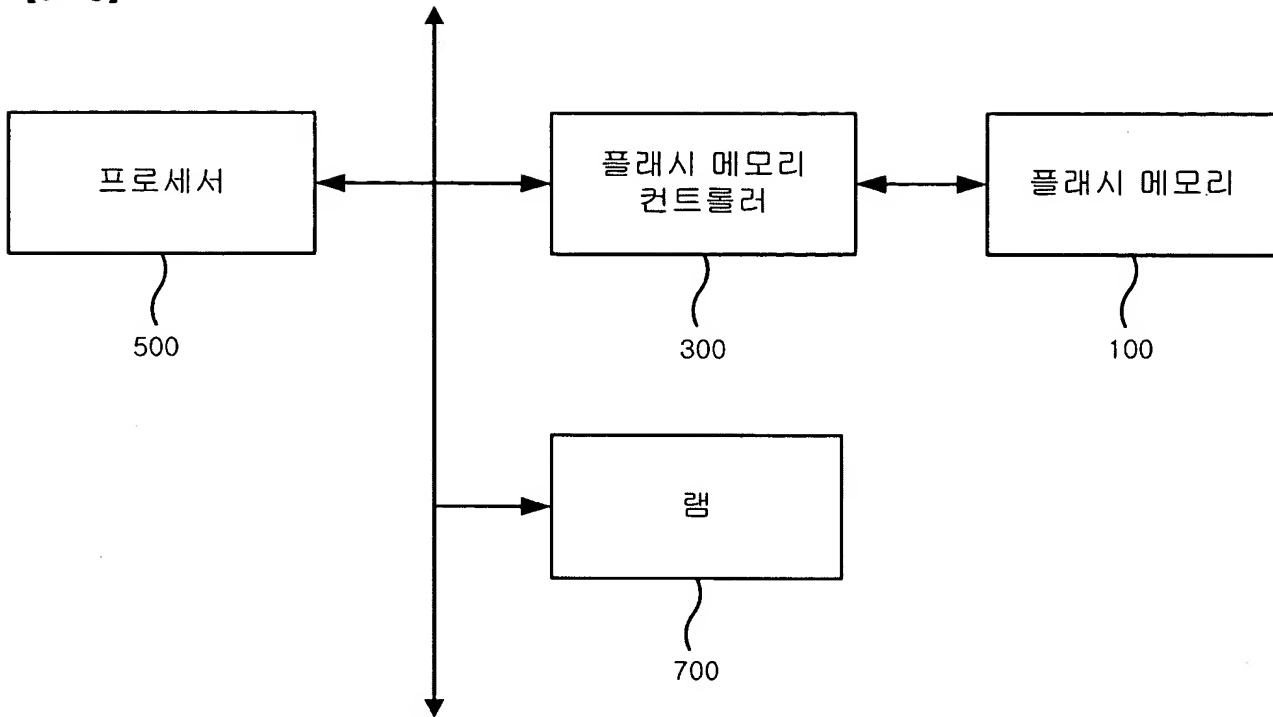
【도 2b】

data	lsn	Cnt	Ecc_lsn	Ecc_data
------	-----	-----	---------	----------

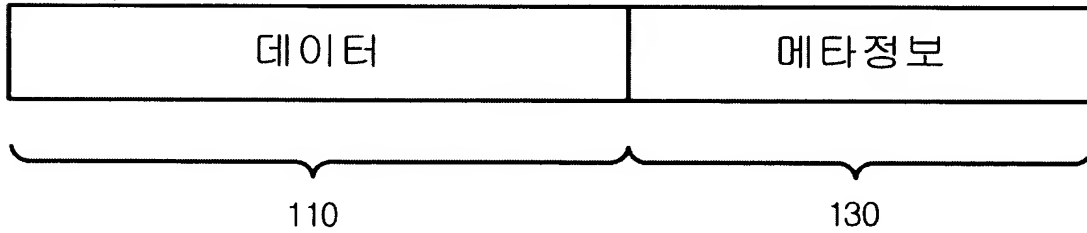
【도 2c】



【도 3】

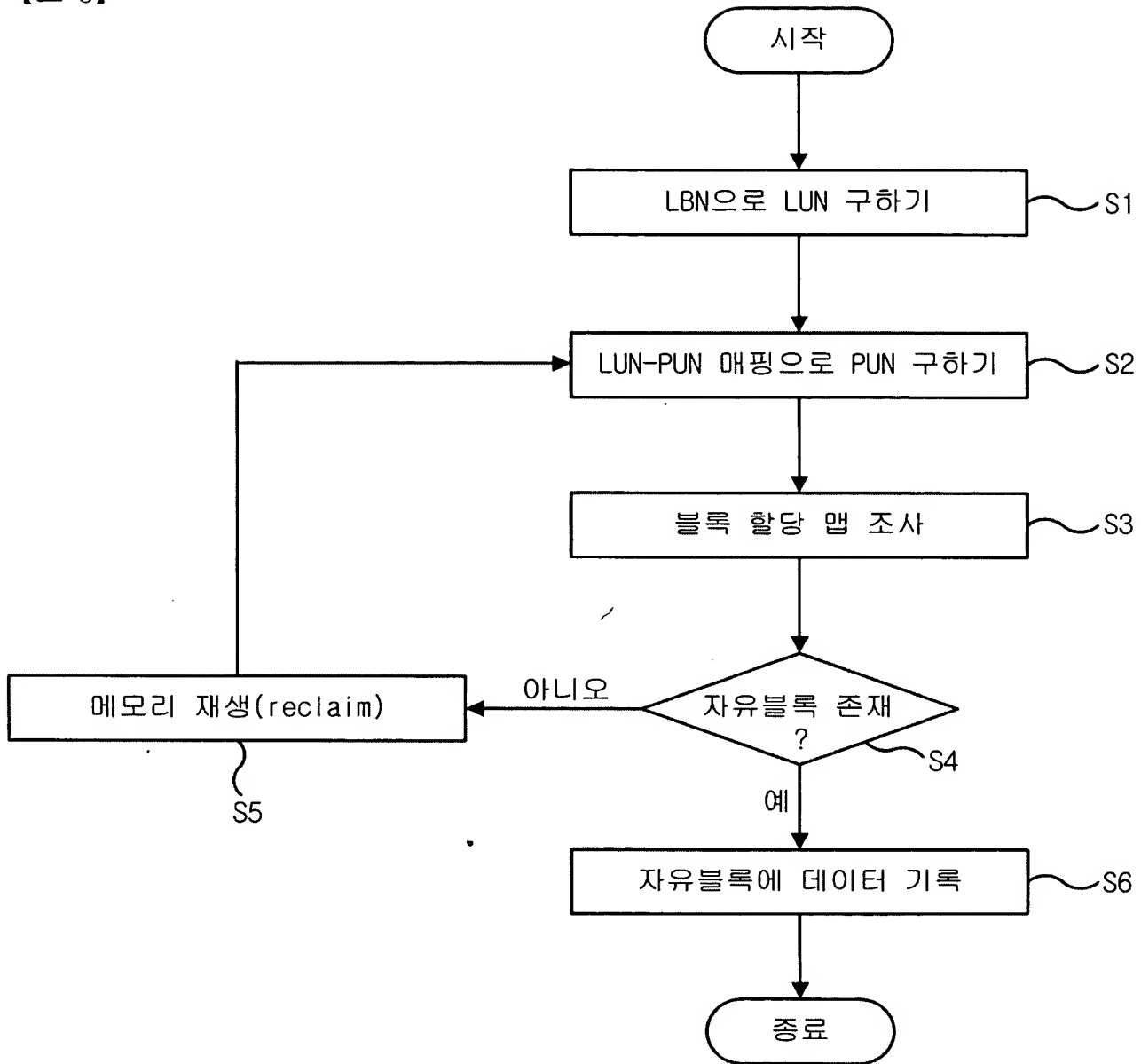


【도 4】

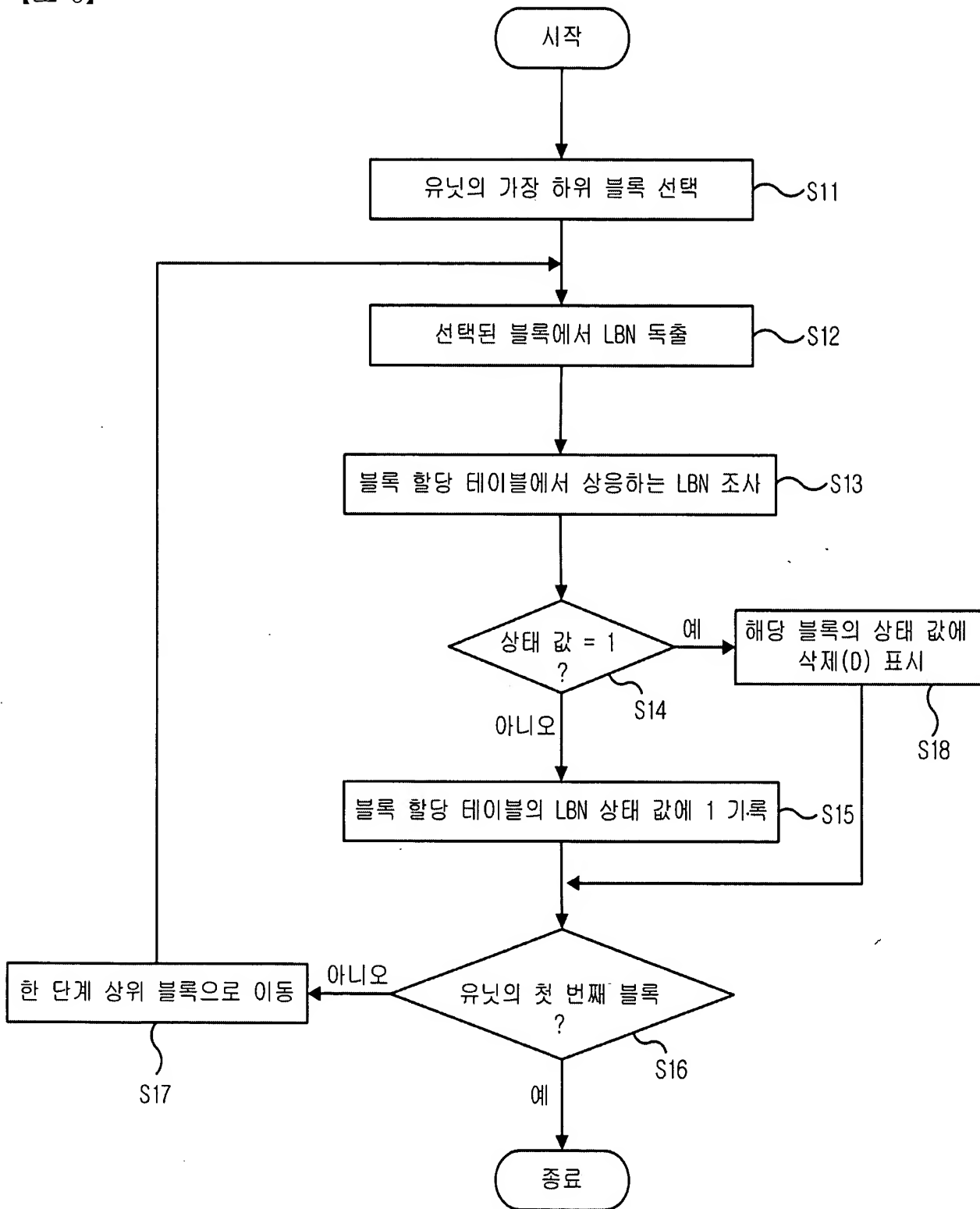




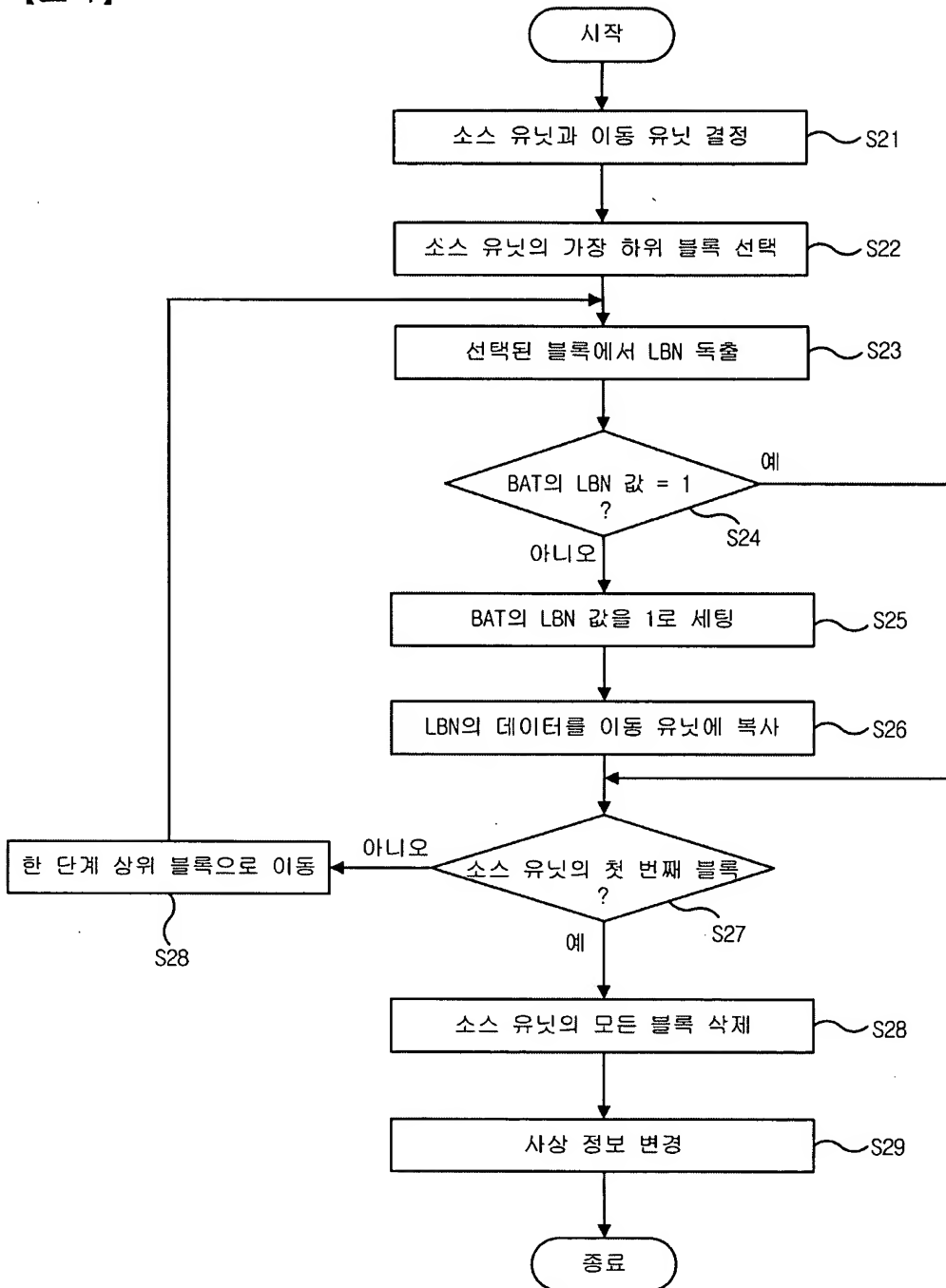
【도 5】



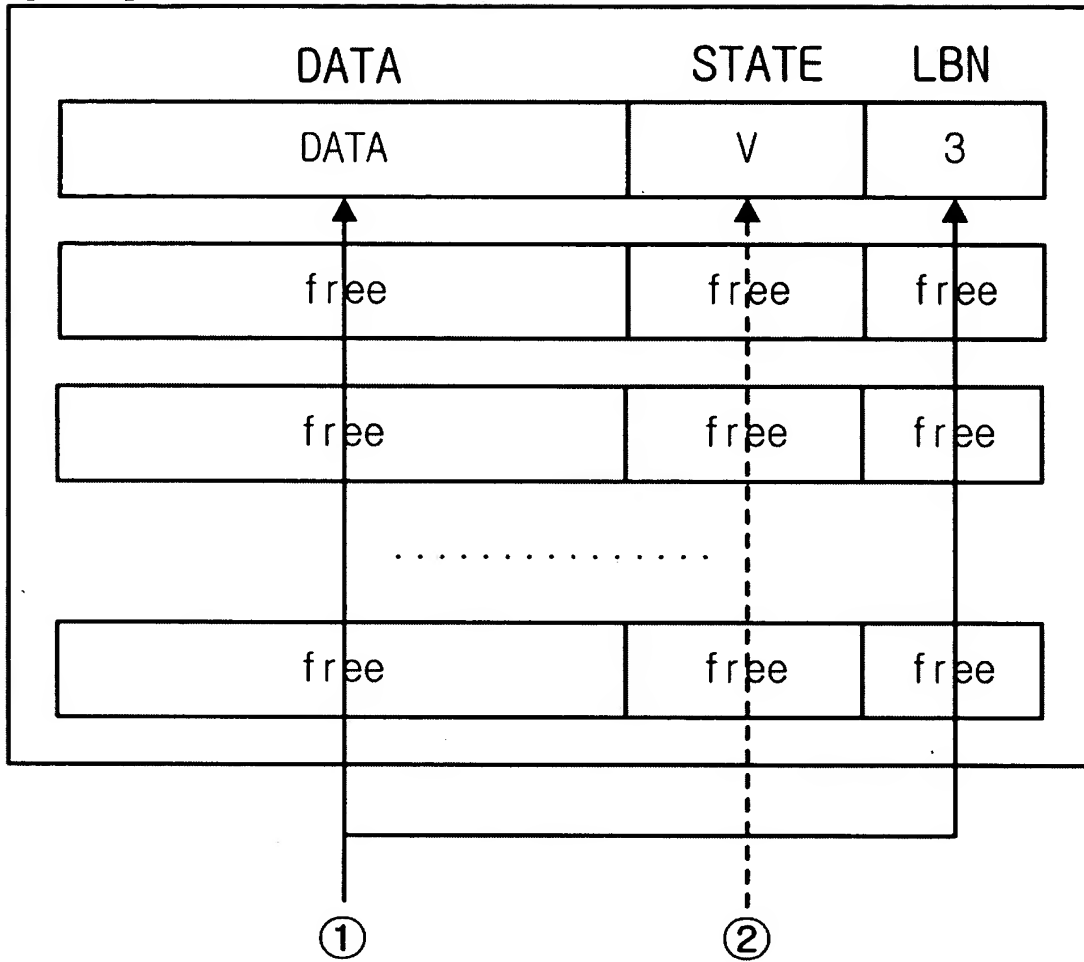
【도 6】



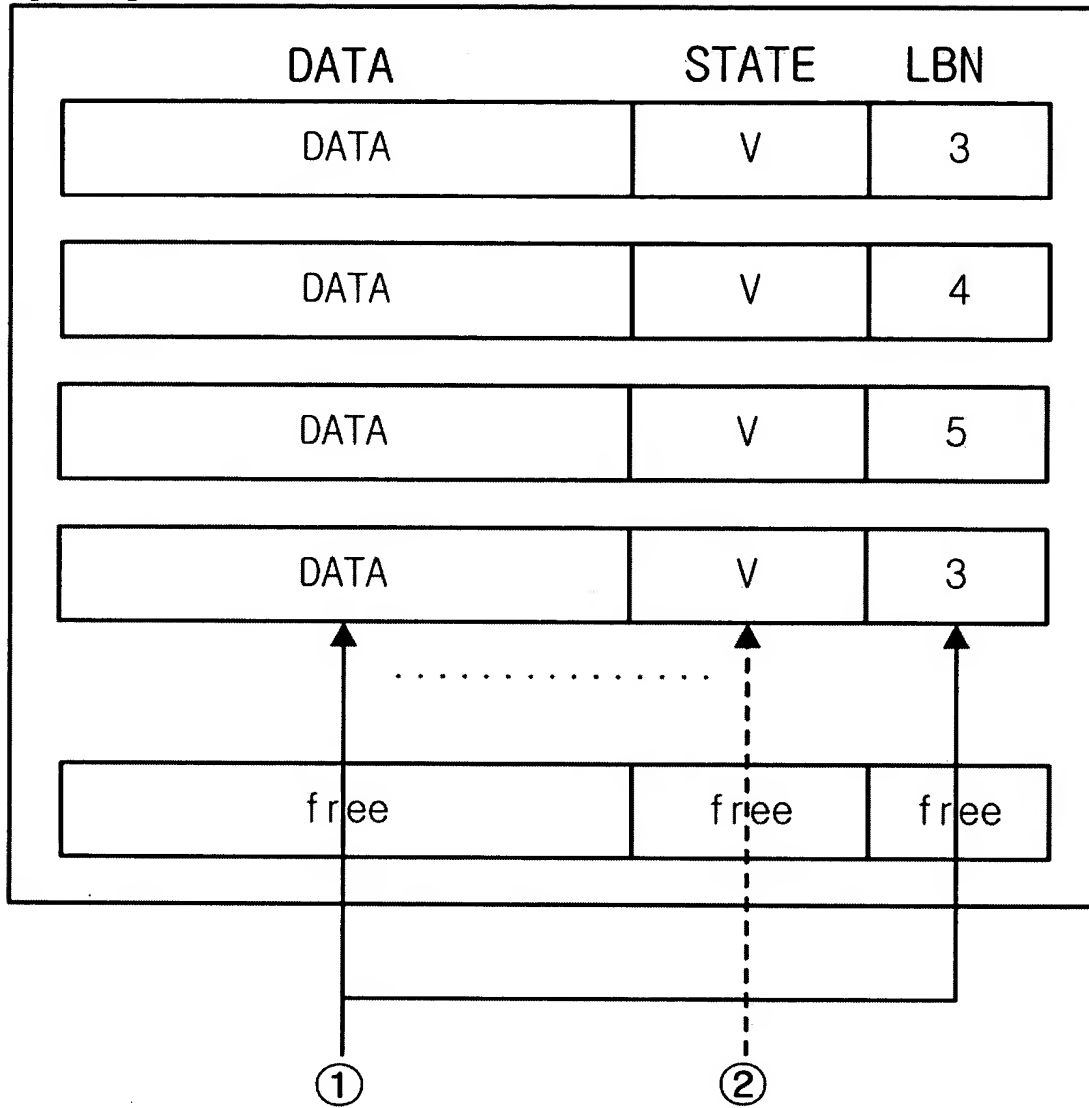
【도 7】



【도 8a】



【도 8b】



【도 8c】

